

UNITED STATES PATENT APPLICATION FOR

A System and Method for Performing Next Placements and Pruning
5 of Disallowed Placements for Programming an Integrated Circuit

Inventors:

Kenneth Ogami

and

10 Rick Hood

Prepared by:

15 WAGNER, MURABITO & HAO, LLP

Two North Market Street

Third Floor

San Jose, California 95113

(408) 938-9060

A System and Method for Performing Next Placements and Pruning of Disallowed Placements for Programming an Integrated Circuit

RELATED APPLICATIONS

5 US Patent Application entitled "MICROCONTROLLER PROGRAMMABLE
SYSTEM ON A CHIP," having docket number CYPR-CD00232, filed on
10/22/2001, and with inventor Warren Snyder is hereby incorporated by
reference.

10 FIELD OF THE INVENTION

The invention relates generally to the field of chip design software
applications, more particularly to a system and method for placing resources
within a chip.

15 BACKGROUND OF THE INVENTION

It is often useful to utilize chip design application software to layout and
plan new chips. This chip design application software is typically configured to
aide the user in keeping track of resource requirements of particular modules.
Furthermore, chip design application software also allows users to assign chip
20 resources to particular modules.

However, chip design software applications typically have minimal
graphical support. They usually are not capable of supplying the user with a

graphical display representing a current status of the layout of the resources on a chip. Chip designers are typically required to manually and textually track their layout decisions with minimal graphical support. Further, typical software packages do not give graphical representations of possible placement of resources for unplaced user modules. Additionally, typical software packages also do not provide automated possible placements for user module resources.

Using the conventional art, a chip designer examines the vacant hardware resources and manually determines which hardware resources can be used for which user modules. This task involves manually determining the set of resources available in a hardware block and comparing them to the resources needed for a user module. This manual test is very technically complex and user-prone. Further, because possible placements require a great deal of manual effort, optimization through iteration trial and error is typically never accomplished.

SUMMARY OF THE INVENTION

It is useful to provide a chip designer with a chip design application software that provides the chip designer with an automated placement of user module resource onto the chip given constraints of the chip resources and the requirements of the user module resources. For example, having a placement of resources for a user module automatically be performed without low level programming by a user would be useful. Further, being able offer alternate placement possibilities for resources of user modules would also be beneficial.

A system and method are described for graphically displaying modules and resources within a chip design software application. The system and method provide a data driven model for matching the hardware resource requirements for an associated user module and the available hardware resources on an underlying chip. In this way, possible placements of a user module can be inferred from the data descriptions of the hardware resources and the user modules. In one embodiment, the data descriptions are formatted using XML data. Databases are utilized to describe the hardware resource requirements which are dictated by the particular user module and the available hardware resources of a particular chip. The user module descriptive database can be updated in response to additional user modules being added or changes to the hardware resource requirements of existing user modules. The hardware description database can be updated in response to additional chips being

added. Further, the graphical interface relates both a user module and the possible hardware resource. This graphical interface utilizes highlights of both the module and the associated resource in patterns, grayscales, or colors to graphically illustrate the relationship between the module and the associated resource.

User modules may require multiple hardware blocks to implement. In some cases, user modules may require special ports or hardware which will limit the number of hardware blocks that can be used for their implementation. The process of mapping hardware blocks to a user module, such that the user module is realized within the microcontroller, is called "user module placement."

Embodiments of the present invention relate to an automatic process that determines the possible placements of a user module based on (1) its XML user module description and (2) the hardware description of the underlying chip. The potential placement positions are automatically inferred based on the XML input data. Therefore, the placement process of the present invention is data driven from this viewpoint.

In one example, when the next placement icon is selected, a potential placement position is computed based on the XML input data. The placement is shown in a graphical hardware layout diagram by highlighting the hardware blocks involved. By clicking the next placement icon, a new placement is then computed

and displayed. Placements that are incompatible with the user module requirements are automatically pruned out. In one embodiment, all positions are shown to the user, sequentially, each time the next placement icon is selected. However, if a potential placement involves a hardware block that has already been
5 used (e.g., by another placed user module), then in these cases the placement icon is grayed out indicating that this placement is only valid if the resources were vacant. This allows the user to see all possible placements.

10 An advantage is that the placement process is data driven based on the XML descriptions of the user modules and hardware. The placements that are computed are inferred based on these descriptions.

15 More specifically, an embodiment of the present invention is drawn to a computer implemented method of determining hardware resources for an electronic design comprising: a) selecting an electronic design represented as a user module; b) accessing a data description of resources required for the user module; c) accessing data descriptions of a plurality of programmable resources of an electronic device; and d) comparing the data description of the user module with the data descriptions of the plurality of programmable resources to
20 automatically determine potential placement options of the user module among the plurality of programmable resources.

Embodiments are also directed to a method as described above and further comprising: displaying on a graphical user interface, a first potential placement of the potential placement options; and in response to a user selecting a next placement icon, displaying on the graphical user interface, a second potential placement of the potential placement options, wherein potential placement options are displayed using visual attributes and wherein the electronic device is a programmable microcontroller device.

Embodiments include the above and wherein the user module requires one or more programmable resource to place and wherein the plurality of programmable resources comprise a plurality of analog programmable resources and a plurality of digital programmable resources.

Embodiments also include the above and wherein the data descriptions are created in XML.

Other aspects and advantages of the invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, and illustrated by way of example of the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a system for coding modules and associated resource(s) in accordance with the invention.

Figures 2A, 2B, and 2C illustrate various embodiments for color coding modules and associated resources in accordance with the invention.

Figure 3 illustrates a process flow diagram of one embodiment of the invention.

Figure 4 illustrates a display screen from one embodiment of the invention.

Figure 5A illustrates an exemplary screen diagram of a next placement iteration procedure of an embodiment of the present invention where one user module is placed.

Figure 5B illustrates an exemplary screen diagram of a next placement iteration procedure of an embodiment of the present invention where a subject user module is selected and showing an initial possible placement designation for the subject user module.

Figure 5C illustrates an exemplary screen diagram of a next placement iteration procedure of an embodiment of the present invention where a subject user module is selected and the digital portion of the initial placement is maintained while the analog portion is iterated to a next placement (second).

Figure 5D illustrates an exemplary screen diagram of a next placement iteration procedure of an embodiment of the present invention where a

subject user module is selected and the analog portion of the second placement is maintained while the digital portion is iterated to a next placement (third).

Figure 5E illustrates an exemplary screen diagram of a next placement

5 iteration procedure of an embodiment of the present invention where a subject user module is placed using the third placement of Figure 5D.

FIG. 5E

DETAILED DESCRIPTION

Specific reference is made in detail to the embodiments of the invention, examples of which are illustrated in the accompanying drawings. While the invention is described in conjunction with the embodiments, it will be understood

5 that the embodiments are not intended to limit the scope of the invention. The various embodiments are intended to illustrate the invention in different applications. Further, specific details are set forth in the embodiments for exemplary purposes and are not intended to limit the scope of the invention. In other instances, well-known methods, procedures, and components have not

10 been described in detail as not to unnecessarily obscure aspects of the invention.

With reference to Figure 1, a system 100 for utilizing a data driven model for matching the hardware resource requirements for an associated user module and the available hardware resources on an underlying chip is shown.

15 Further, the system 100 graphically illustrates alternative possible placements for user module resources and automatically generates alternative placements for user module resources based on the requirements of the user module and the resource availability on the underlying chip. The system 100 operates within a chip design application to automatically generate possible placements for user

20 module resources based on the requirements of the user module and the resource availability on the chip. Further, the system 100 also operates to graphically display the correlation between an unplaced module and multiple alternate possible resources associated with the unplaced module. In one

embodiment, the graphical correlation between the unplaced module and the possible resources associated with the unplaced module are displayed by use of a corresponding color within the design application.

5 The system 100 includes a processor 140, a user input interface 130 (e.g., cursor control device and keyboard), volatile memory 150, a video processor 160, and non-volatile memory 170. The input interface 130, the volatile memory 150, the video processor 160, and the non-volatile memory 170 are connected to the processor 140. The input interface 130, the processor 140, the volatile
10 memory 150, the video processor 160, and the non-volatile memory 170 are components that are readily found in personal computer systems.

 The system 100 further includes a user module description database 110, a resource placement locator 120, a color coordinator 180, and a hardware
15 description database 185, which are also connected to the processor 140. The components 110, 120, 180, and 185 are merely illustrated in Figure 1 as one embodiment of the system 100. Although the components 110, 120, 180, and 185 are illustrated in Figure 1 as separate components of the system 100, two or more of these components may be integrated, thus decreasing the number of
20 components in the system 100. Similarly, the components 110, 120, 180, and 185 may also be separated, thus increasing the number of components within the system 100. The components 110, 120, 180, and 185 may be implemented in any combination of hardware, firmware and software.

In one embodiment, the system 100 helps users more accurately and efficiently design chip layouts. The system 100 automatically finds potential placements of resources which fulfill the requirements of the associated user
5 module.

The system 100 can iterate through multiple potential placement possibilities for resources, thus giving the user of the system 100 multiple choices for resource placement.

Further, the system 100 also graphically displays relationships between the user module and the potential placement locations for the resources. The system 100 can also graphically display locations that are currently occupied by another user module but otherwise could have been a potential placement
10 location.
15

In one embodiment, the system 100 is configured to support microcontroller design. In another embodiment, the system 100 is configured to support programmable system on a chip design. In yet another embodiment, the
20 system 100 supports general chip design.

The input interface 165 provides a means for the system 100 to receive user input which may include selection of various user module and resources

and command sequences. The input interface 165 may be a USB port, a serial port, Ethernet port, or any other interface port configured to transmit electronic data to the system 100.

5 The video processor 160 provides graphical output from the system 100. The video processor 160 is configured to display color coded user modules and corresponding resources.

10 The user module description database 110 contains descriptions of the required hardware resources needed by a particular user module. This information may be stored using XML data. In addition to a list of hardware resources that are needed, the user module description database 110 also stores the specific configuration requirements of the needed hardware resources. Some of the description of the required hardware resources contains detailed
15 configuration parameters such as pin restrictions, resource dependencies, speed requirements, and the like. For example, due to communication requirements between hardware resources, these hardware resources may need to be located in close proximity to each other. Another example, due to performance requirements, certain hardware resources may need to be located in close
20 proximity to each other.

In one embodiment, the user module description database 110 contains descriptions of hardware resources within a user module. The user module

description database 110 can include the hardware resource requirements of many different user modules. In this embodiment, the user module description database 110 would be applicable across a plurality of underlying chips. The contents of the user module description database 110 can be updated based on changes to the resource requirements of the user module or the introduction of new user modules not currently contained within the user module description database 110. In one embodiment, the user module description database 110 is updated from an outside source. The updated data can be routed through the input interface 130. In one embodiment, the information within the user module description database 110 can also be stored within the volatile memory 150 and/or the non-volatile memory 170.

In one embodiment, the user module description database 110 is implemented in XML. In another embodiment, the user module description database 110 is implemented in any other mark-up language.

The hardware description database 185 contains descriptions of hardware resources within the underlying chip. These descriptions includes various attributes of the hardware resources such as the functionality of the resources, the interconnectivity between these resources, the operating parameters of the resources, the pin layouts of the resources, and the like.

In one embodiment, the hardware description database 185 contains descriptions of hardware resources for multiple underlying chips. In this embodiment, the hardware description database 185 would be applicable across a plurality of underlying chips. The contents of the hardware description database 185 can be updated based on changes to the resources within the underlying chip or the introduction of new chips not currently contained within the hardware description database 185. In one embodiment, the hardware description database 185 is updated from an outside source. The updated data can be routed through the input interface 130. In one embodiment, the information within the hardware description database 185 can also be stored within the volatile memory 150 and/or the non-volatile memory 170.

In one embodiment, each instance of a change in utilized resources within the underlying chip triggers an update within the hardware description database 185. For example, when a hardware resource changes from being utilized to being available because the associated placed user module become "unplaced", then the hardware description database 185 is updated with the newly available hardware resources. Likewise, when a hardware resource changes from being available to being utilized because the associated user module is "placed", then the hardware description database 185 is updated with the newly unavailable hardware resources.

In one embodiment, the hardware description database 185 is implemented in XML. In another embodiment, the hardware description database 185 is implemented in any general database format which is compatible with the database engine used.

5

The resource placement locator 120 locates available hardware resources on an underlying chip that would be suitable for realizing an unplaced module. The module, or user module is a circuit design. In one embodiment, the resource placement locator 120 is configured to accept the hardware resource requirements for the unplaced module from the user module description database 110 and to search for a resource from the available resources that would satisfy these requirements from the hardware description database 185. The resource placement locator 120 can utilize the information describing the hardware resource requirements of a user module and find a suitable match based on that information. Hardware resources on the underlying chip which are incompatible with the user module are automatically disregarded and pruned out from the selection of suitable resources.

In one embodiment, the hardware resources that are currently utilized by another user module which would otherwise be suitable for a current user module are grayed out indicating that placement of these resources would only be valid if the resources were vacant. In another embodiment, occupied

hardware resources would not be highlighted and would be disregarded and pruned.

In another embodiment, the resource placement locator 120 sequentially
5 searches for possible resource configurations from the available resources. For
example, the resource placement locator 120 can be configured to find a first set
of resources which fulfill the requirements for the unplaced module. Next, the
resource placement locator 120 can be configured to sequentially find a second
set of resources that are different from the first set of resources which also fulfill
10 the requirements for the unplaced module.

The color coordinator 180 graphically matches the module and the
associated corresponding resources. In one embodiment, the color coordinator
180 color codes the module and the associated corresponding resources. In one
15 embodiment, the color coordinator 180 is configured to select a unique color to
display both an unplaced module and a possible set of available resources
corresponding to the requirements of the unplaced module. In another
embodiment, the color coordinator 180 is configured to select a unique color to
display an unplaced module and another unique color to display a fixed resource
20 and another unique color to display a next placement resource.

In one embodiment, matching colors can be utilized. In another embodiment, matching grayscales also can be utilized. In yet another embodiment, matching patterns can also be utilized.

5 Figures 2A, 2B, and 2C each illustrate one embodiment of the color coordinator 180 displaying a unique color that corresponds with a module and resources which correspond with the module. For the sake of clarity, common element numbers are utilized to represent similar items to avoid unnecessary confusion. For example, a module 210 and the corresponding resources 220 and 230 are utilized in Figures 2A, 2B, and 2C to merely illustrate the different
10 embodiments of color coding the module 210 with the corresponding resources 220 and 230. Additional modules and resources can be displayed simultaneously.

15 In Figure 2A, a ring 235 appears around an icon representation of the module 210. In one embodiment, the ring 235 is displayed filled in with a cross-hatched pattern 240 to represent a unique color. However, in other embodiments, different shading techniques may be utilized. The resources 220 and 230 are also filled in with the cross-hatched pattern 240. The same cross-
20 hatched pattern 240 within the ring 235 and within the corresponding resources 220 and 230 visually indicate that the module 210 corresponds to the resources 220 and 230.

In Figure 2B, the module icon 210 is displayed filled in with a cross-hatched pattern 245 to represent a unique color. However, in other embodiments, different shading techniques may be utilized. The resources 220 and 230 are also filled in with the cross-hatched pattern 245. The same cross-hatched pattern 245 within the module 210 and within the corresponding resources 220 and 230 visually indicate that the module 210 corresponds to the resources 220 and 230.

In Figure 2C, a ring 250 appears around the module icon 210. In one embodiment, the ring 250 is displayed filled in with a cross-hatched pattern 255 to represent a unique color. However, in other embodiments, different shading techniques may be utilized. An area 260 is also filled in with the cross-hatched pattern 255. The area 260 includes the resources 220 and 230. The same cross-hatched pattern 255 within the ring 250 and within the area 260 visually indicate that the module 210 corresponds to the resources 220 and 230.

Figures 3 illustrates a process flow diagram in accordance with one embodiment of the invention. The functional blocks are not to be construed as limiting the number of functional blocks within the process flow diagrams nor to be construed as a requirement for every functional block. The blocks may be performed in a different sequence without departing from the spirit of the invention. Further, blocks may be deleted, added or combined without departing from the spirit of the invention.

Figure 3 illustrates one embodiment showing the selection of an unplaced module and the viable options of possible resources which meet the requirement of the unplaced module. In Block 310, an unplaced module is selected.

5

In Block 320, a description of the required hardware resources associated with the selected unplaced module are located. In one embodiment, the function within the Block 320 can be performed by the user module description database 110 (Figure 1).

10

In Block 330, a description of the underlying hardware resources within the underlying chip are located. In one embodiment, the function within the Block 330 can be performed by the hardware description database 185 (Figure 1).

15

In Block 340, a comparison between the description of the required hardware resources associated with the selected unplaced module and the description of the hardware resources belonging to the underlying chip occur. The result of this comparison is a group of possible hardware resources that satisfy the requirements of the selected unplaced module. In one embodiment, the function within the Block 340 can be performed by the resource placement locator 120 (Figure 1).

20

For instance, if a user module requires a special port, then any hardware resource block not having the port is automatically pruned out of the list by performing the database comparison function of Step 340. Furthermore, if a user module requires multiple hardware resources that need to be adjacent, then any set of hardware resources not meeting this requirement will be automatically pruned out of consideration. In one embodiment, occupied hardware resources are also pruned out. In another embodiment, they are left in to give the user maximum potential placement information. The pruning process is data driven according to the XML databases which are compared to determine the list of possible placement options.

By automatically selecting the possible placements, and automatically pruning the disallowed placements, the user need only click the next placement icon to view the potential placements available to select the optimal placement location for a user module.

In Block 350, the hardware resources of the underlying chip which satisfy the requirements of the selected unplaced module are highlighted. In one embodiment, the hardware resources that are currently utilized by another user module which would otherwise be suitable for a current user module are grayed out indicating that placement of these resources would only be valid if the resources were vacant. In Block 360, the next set of hardware resources of the

underlying chip which satisfy the requirements of the selected unplaced module are highlighted.

Figure 4 illustrates one embodiment of a display screen showing a group of modules and a group of resources. For example, a module grouping 410 and a resource grouping 430 are utilized in Figure 4 to merely illustrate a graphical representation of the general layout of the plurality of modules and resources. Additional modules and resources can be displayed simultaneously.

In one embodiment, Figure 4 illustrates a highlighted module 415 within the module grouping 410. The highlighted module 415 is shown with a ring 420 surrounding the module 415. The ring 420 is shown with a first cross-hatched pattern. The highlighted module 415 graphically illustrates that this particular module is selected from the module grouping 410.

Resources 435, 440, and 450 are shown highlighted and correspond to the module 415. The resources 435, 440, and 450 are shown within the resource grouping 430. The resources 435 and 440 are also shown highlighted with a second cross-hatched pattern 445. The resource 450 is shown highlighted with a third cross-hatched pattern 455.

In one embodiment, the resources 435 and 440 are decoupled from the resource 450 as illustrated by the second cross-hatched pattern 445 and the third

cross-hatched pattern 455, respectively. In one embodiment, the resources 435 and 440 are coupled together and placed as a group.

In one embodiment, the second cross-hatched pattern 445 graphically
5 represent the area covered by the unfixed resources, and the third cross-hatched
pattern 455 graphically represents the area covered by the fixed resources.
Accordingly, in this embodiment, the resources 435 and 440 are initially unfixed,
and the resource 450 is initially fixed. However, the resources 435 and 440 can
become fixed resources at any time by finalizing placement of the resources 435,
10 440, and 450 of the module 415 or by selecting the resource 450 as the unfixed
resource.

In operation, as a next placement is requested, the resources 435 and 440
are iterated to a next available position for placement. The second cross-
15 hatched pattern 445 follows the resources 435 and 440 to their next location. If a
next placement is requested again, the resources 435 and 440 would be iterated
again to the next available position as long as the resources 435 and 440 are
unfixed. At any time during this process, the resources 435 and 440 can have
their placements finalized by either finalizing placement for the resources 435,
20 440, and 450 or by selecting the resource 450 as the unfixed resource.

In another embodiment, there can be more or fewer resources associated
with the second and third cross-hatched patterns 445 and 455. There can also

be more than one group of fixed resources. The second and third cross-hatched patterns 445 and 455 and their associated resources are shown for exemplary purposes.

5 NEXT PLACEMENT ITERATOR EXAMPLE

Figure 5A illustrates an example computer screen diagram 510 of a next placement iterator process in accordance with one embodiment of the present invention. In accordance with the graphical user interface, the digital resources (here, eight) are shown in an upper horizontal row 505 and the analog resources (here, twelve) are shown in a lower situated matrix 507. A selection bar 505 comprises user module icons that can be selected. The user module icon 515 ("counter") is currently selected. The allocated resources 509 that are designated to implement user module 515 are also highlighted. In this embodiment, the color ring that surrounds user module icon 515 is color coded to the allocated resources 509. Therefore, this user module 515 is currently placed. The remaining user module icons of the selection bar 504 remain unplaced.

Figure 5B illustrates an example computer screen diagram 520 of the next placement iterator process in accordance with one embodiment of the present invention where the user selects an unplaced user module icon 525 (the "ADCINC"). Since the module 525 is unplaced, it does not have an associated color ring. Upon selection of the user module icon 525, an initial possible placement for this design is displayed. The initial possible placement includes

two digital resources (blocks) 530a and one analog resource 530b. In the embodiment shown, only vacant blocks were selected as the initial placement, however, in another embodiment, the computer could also designate a used block as a potential placement option for user module 525. Of course, a block
5 would have to be made vacant before it could be used for user module 525.

Figure 5C illustrates an example computer screen diagram 530 of the next placement iterator process in accordance with one embodiment of the present invention where the user invokes a next placement iteration for module icon 525 (the "ADCINC"). In particular, the user uses the cursor control device to select
10 resource 530b. This causes the cross hatching behind the analog resource 530b to change colors from the cross hatching behind the digital resources 520a. Once selected, the user clicks the "next placement" icon 590, this causes the analog resource to move from its initial location in
15 Figure 5B to its new location in Figure 5C. Figure 5C therefore illustrates a second possible placement for the selected user module 525. By selecting the analog resource 530b before pressing the next placement icon 590, the user decoupled the placement of the digital versus analog resources. In other words, the digital resources 530a remained fixed from Figure 5B to
20 Figure 5C.

Figure 5D illustrates an example computer screen diagram 540 of the next placement iterator process in accordance with one embodiment of the present

invention where the user invokes a next placement iteration for module icon 525 (the "ADCINC"). In particular, the user uses the cursor control device to select digital resource 530a. This causes the cross hatching behind the digital resources 530a to change colors from the cross hatching behind the analog resource 520b. Once selected, the user clicks the "next placement" icon 590, this causes the digital resource 530a to move from its initial location in Figure 5B to position 509 (an occupied position). The user clicks the icon 590 again thereby causing the digital resource 530a to appear in its position as shown in Figure 5D. Figure 5D therefore illustrates a third possible placement for the selected user module 525. By selecting the digital resource 530a before pressing the next placement icon 590, the user decoupled the placement of the digital versus analog resources. In other words, the analog resource 530b remained fixed from Figure 5C to Figure 5D.

Figure 5E illustrates an example computer screen diagram 560 of the next placement iterator process in accordance with one embodiment of the present invention where the user then places the user module 525. In accordance with the graphical user interface, the user then selects the "place user module" icon 595 and the user module 525 becomes placed using the last possible placement. In accordance with placing, a color ring appears around the module icon 525. Further, the hardware resources 530 appear in a matching color and they now have labels ("ADCINC . . .") that correspond to the placed icon 525.

By decoupling the digital from the analog resources during the next placement iteration process, the present invention reduces the number of possible placements that have to be cycled through by the user before the desired placement is found.

5

The foregoing descriptions of specific embodiments of the invention have been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise embodiments disclosed, and naturally many modifications and variations are possible in light of the above teaching. The embodiments were chosen and described in order to explain the principles of the invention and its practical application, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the

10

15

Claims appended hereto and their equivalents.